



Reconocida internacionalmente por la acreditadora CQAIE ( Washington, USA)

**UAI** Universidad Abierta  
Interamericana

**UAIOnline**

**Evaluación Parcial**

Carrera: Analista Programador

**Asignatura:** PROGRAMACIÓN ORIENTADA A OBJETOS

## Evaluación Parcial N°2

### Propuesta de Integración del Módulo II

Nombre y apellido:	
DNI:	
Unidad de Apoyo:	

#### Presentación

El trabajo que inicia constituye su primera evaluación parcial de la asignatura. Como tal, su realización es obligatoria y constituye una condición para rendir el examen final.

Cada ítem debe ir acompañado de todos aquellos comentarios referentes a dificultades y aclaraciones que permitan una mayor comprensión del proceso de resolución de cada consigna.

Esta propuesta de trabajo procura orientarlo/a para que se acerque a las siguientes metas de aprendizaje:

Profundizar los conocimientos sobre la programación orientada a objetos.  
Ejercitar la escritura de programas en el entorno VS.NET

Tenga en cuenta que todos los contenidos abordados en este módulo (Unidades 4, 5 y 6) más los vistos previamente son fundamentales para la realización de esta Propuesta de Integración Modular.

Le pedimos que Desarrolle el programa en C# con el visual studio sugerido en el curso. Documentar dentro del código todo lo necesario para interpretar lo que quiso programar, **se recomienda que coloque un comentario cada 3 o 4 líneas de código.**

Terminado el programa comprima la carpeta que lo contiene con todos los archivos y entréguelo en el lugar consignado.

El nombre del archivo comprimido debe indicar el Apellido del alumno, la asignatura y el número de parcial, p.e. PEREZ\_POO\_2P.

A continuación le presentamos las consignas de trabajo. Recuerde consultar el Cronograma de Actividades para agendar la fecha de entrega.



## Consignas

**NOTA: RESPETAR LAS BUENAS PRÁCTICAS QUE PROPONE LA ORIENTACIÓN A OBJETOS. ES MUY IMPORTANTE DETECTAR LAS ABSTRACCIONES Y LAS RELACIONES QUE EXISTEN ENTRE ELLAS. LO EXPRESADO ES MUY IMPORTANTE.**

Una empresa que se dedica a comercializar artículos de plástico nos encarga el desarrollo de un programa que les permita conocer el stock y costo que poseemos de cada uno de ellos.

Los artículos se identifican por un código alfanumérico que posee una letra mayúsculas y 5 dígitos (p.e P00001). Al ingresar estos datos se deben validar ara que cumpla con lo requerido y no exista un código repetido.

También los artículos poseen: descripción, costo, fecha de alta (que se carga cuando el artículo entra en vigencia (Alta)), fecha de baja (que se asigna cuando un artículo deja de comercializarse). Si el artículo se ha dejado de comercializar no se pueden realizar cambios en él ni en su stock.

Nos solicitan el ABM de estos productos. También que el sistema tenga la posibilidad de aumentar el stock o disminuirlo. Un artículo no puede poseer stock negativo. Para cada operación colocar un botón (alta, baja, modificación, agregar stock, disminuir stock).

Para determinar el costo de un artículo nos hacen expresa mención que la forma de cálculo es parte de un artículo. La misma no cambia durante todo el ciclo de vida del artículo y se crea cuando él se instancia. Existen tres formas de cálculo diferentes que se detallan a continuación. Además, cada forma de determinación de costo posee un id y una descripción.

La primera forma determina un costo promedio ponderado CPP. La forma de cálculo se establece de la siguiente manera:

CPP = Costo promedio ponderado actual.

STA = Stock actual del artículo.

CUA = Costo unitario de los artículos adquiridos que ingresan al stock.

QA = Cantidad de artículos adquiridos.

Nuevo costo de este artículo =  $((CPP * STA) + (CUA * QA)) / (STA/QA)$

La segunda forma denominada costo último entrado CUE, toma como costo del artículo para todo el stock de ese artículo, el costo unitario del último lote de ingresado.

La tercera forma se denomina costo personalizado CP. La forma de cálculo se establece de la siguiente manera:

CP = Costo personalizado actual del artículo.

CA = Costo del artículo adquirido que ingresa al stock.

Nuevo costo de este artículo =  $((CP + CA) / 2) * 1,1$

El costo de un determinado artículo debe actualizarse cada vez que ingresan más de ese mismo tipo.

Cuando se da de alta un artículo nuevo, el costo unitario siempre es el de esa compra.



Los datos se deben mostrar en una grilla DataGridView. Los datos a visualizar en la grilla para cada artículo son:

1. Código del artículo.
2. Descripción.
3. Fecha de alta
4. Fecha de baja
5. Meses vigentes (\*)
6. Costo
7. Descripción del método de costo utilizado
8. Stock

(\*) La cantidad de meses entre la fecha de baja y la de alta. Validar para que una fecha de alta nunca sea posterior a una de baja. Si el artículo aún está vigente pues no posee fecha de baja, en “meses vigente” mostrar “--”).

El sistema posee un evento denominados StockCero que se desencadena cuando un stock llega a cero. El evento lleva un parámetro personalizado con la información del código y descripción del artículo que provocó el desencadenamiento del evento.

Además, nos solicitan que coloquemos botones para solucionar las siguientes funcionalidades y que los resultados se muestren en una segunda grilla.

- a. Utilizando Linq proyectar todos los artículos cuyo stock sea superior al valor ingresado por el usuario.
- b. Utilizando LinQ proyectar la descripción de todos los artículos dónde el stock sea igual a cero.
- c. Utilizando Icomparer mostrar todos los artículos ordenados alfabéticamente por descripción de manera ascendente y descendente.
- d. Utilizando LinQ e Icomparer proyectar todos los artículos que poseen fecha de baja ordenados de mayor a menor por la cantidad de meses vigente.
- e. Utilizando expresiones lambda muestre en la grilla todos los artículos cuyo Stock se encuentre por sobre el valor n1 ingresado por el usuario y por debajo del valor n2 también ingresado por el usuario. (verifique que n2 sea superior a n1).

También se debe mostrar en una caja de texto el dinero inmovilizado total que tenemos en el stock (cantidad \* costo de todos los artículos). Este valor se debe actualizar cada vez que realicemos una operación que lo afecta.

Todo el código deberá ser controlado por Try... Catch y aplicar excepciones personalizadas.

Coloque todos los elementos en un solo formulario que oficia de GUI.

Todos los valores ingresados deben ser validados y no se deben permitir el ingreso de valores inconsistentes (p.e alta de cantidades negativas, códigos repetidos etc).

Utilice LinQ (de manera explicita como se muestra en el orientador from .. where .. select) y expresiones Lambda para filtrar las grillas.

La buena usabilidad del software por medio de la GUI es parte de la evaluación.



Reconocida internacionalmente por la acreditadora CQAIE ( Washington, USA)

**UAI** Universidad Abierta  
Interamericana

**UAIOnline**

**Evaluación Parcial**

---

## Criterios de evaluación

---

*La presentación de los criterios de corrección pretende servirle como una herramienta para que usted anticipe los logros obtenidos y ajuste su producción a los requerimientos de la tarea.*

Para la evaluación de este trabajo tendremos en cuenta los siguientes criterios de evaluación:

- Entrega en tiempo y forma.
- Claridad conceptual al aplicar las técnicas y elementos estudiados.
- Respetar las consignas del enunciado al momento de desarrollar la solución.

*Envíe su producción a su tutor/a a través de la herramienta **ENTREGA DE EXÁMENES PARCIALES** que le ofrece el aula virtual.*

*Consulte la fecha de entrega en el Cronograma de la Asignatura. En el término de 15 días usted recibirá una devolución de su docente, a la que podrá acceder consultando sus **CALIFICACIONES**.*

*Recuerde que este trabajo es su Segundo Examen Parcial de la Asignatura.  
Su realización y aprobación es requisito para el acceso al examen final de la asignatura.  
Cumpla con los tiempos y las formas establecidas.  
Gracias y éxitos!*